The Counter-Intuitive Properties of Ensembles for Machine Learning

or

Democracy Defeats Meritocracy

Philip Kegelmeyer, Sandia National Labs, wpk@sandia.gov

(Slides at: www.ca.sandia.gov/avatar)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

DEAN Seminar, April 2, 2008

Conclusions (version 1.0)

If you use **supervised machine learning**, use **ensembles**.

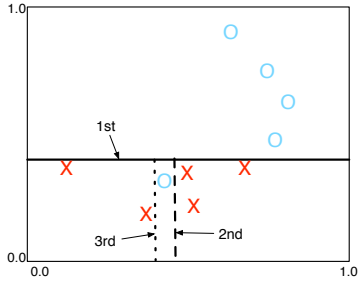
Invented Training Data, for Search Relevance

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_3	No	3	27	0.12	...	0.13
q_4	Yes	16	183	0.08	...	0.58
q_5	No	17	665	0.36	...	0.64
q_6	No	44	1212	0.29	...	0.42
q_7	No	42	24	0.33	...	0.88
q_8	Yes	78	42	0.44	...	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	12	3141	0.92	...	0.17

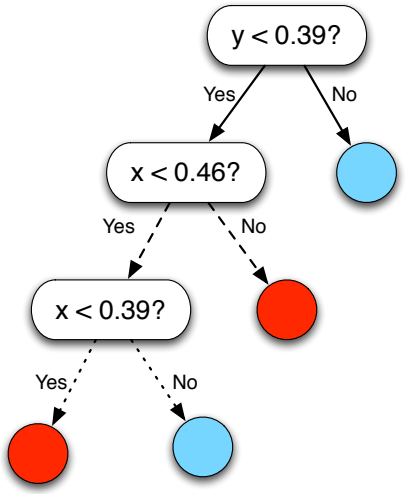
Supervised Machine Learning Overview

Also known as: pattern recognition, statistical inference, data mining.

- Input: "ground truth" data.
  - Samples, with attributes, and *labels*.
  - Example: search result data
    - Samples: a query string
    - Attributes: features of the search
    - Labels: "relevant", "irrelevant"
- Apply suitable method:
  decision trees, neural nets, SVMs.
- Output:
  rules for labeling new, *unlabeled* data.
  Equivalently:
  a partitioning of attribute space.



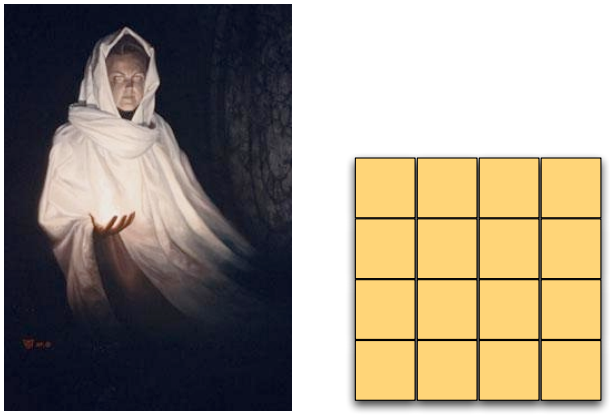
Attribute space partitioned.



Decision tree representation.

Machine Learning, Before Ensembles

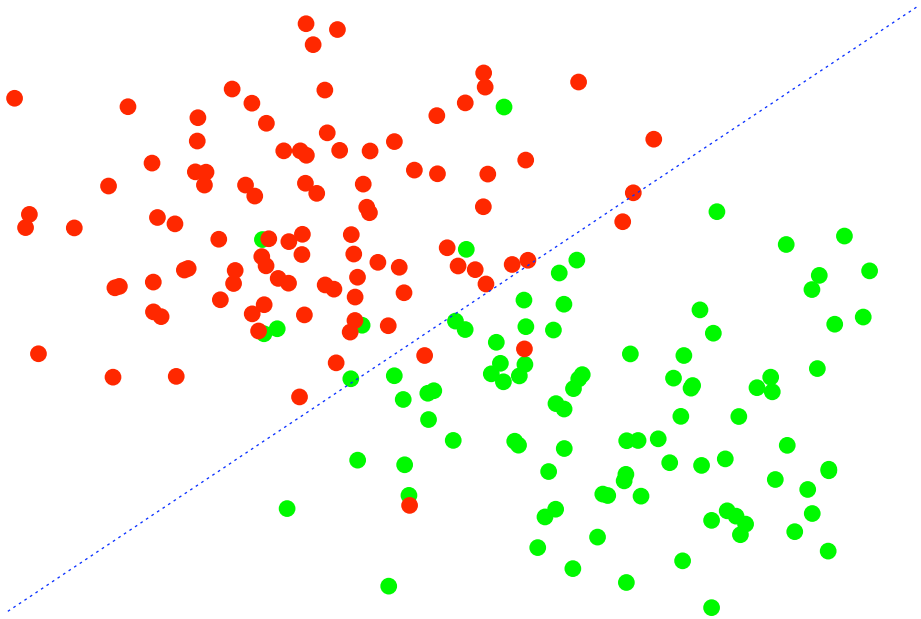
Traditional: Use 100% of training data to build a sage.



Sage sees all the data.

Note: Even Sage is Not Perfectly Accurate

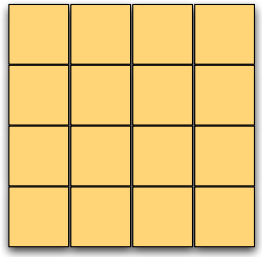
Class distributions can overlap inextricably.



"Bayes error" is the best any classifer can do.

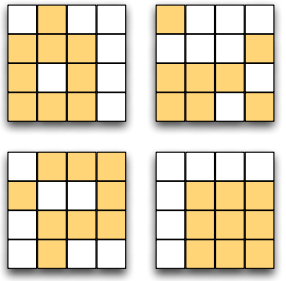
Machine Learning, With Ensembles

Traditional: Use 100% of training data to build a sage.



Sage sees all the data.

Ensembles: Use randomized 100% of training data to build an expert. Repeat to build many experts. Vote them.



Each expert sees 2/3rds of the data.

The experts beat the sage[1]!

Reminder: The Unaltered Training Data

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_3	No	3	27	0.12	...	0.13
q_4	Yes	16	183	0.08	...	0.58
q_5	No	17	665	0.36	...	0.64
q_6	No	44	1212	0.29	...	0.42
q_7	No	42	24	0.33	...	0.88
q_8	Yes	78	42	0.44	...	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	12	3141	0.92	...	0.17

First Expert Sees A Sampling With Replacement

Queries	Relevant? Truth	PageRank a_1	Fresh? a_2	Unique? a_3	... ...	Distinct? a_K
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_2	Yes	99	2	0.33	...	0.03
q_4	Yes	16	183	0.08	...	0.58
q_4	Yes	16	183	0.08	...	0.58
q_5	No	17	665	0.36	...	0.64
q_8	Yes	78	42	0.44	...	0.52
q_9	No	59	7012	0.37	...	0.23
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_{N-1}	Yes	36	1812	0.47	...	0.17

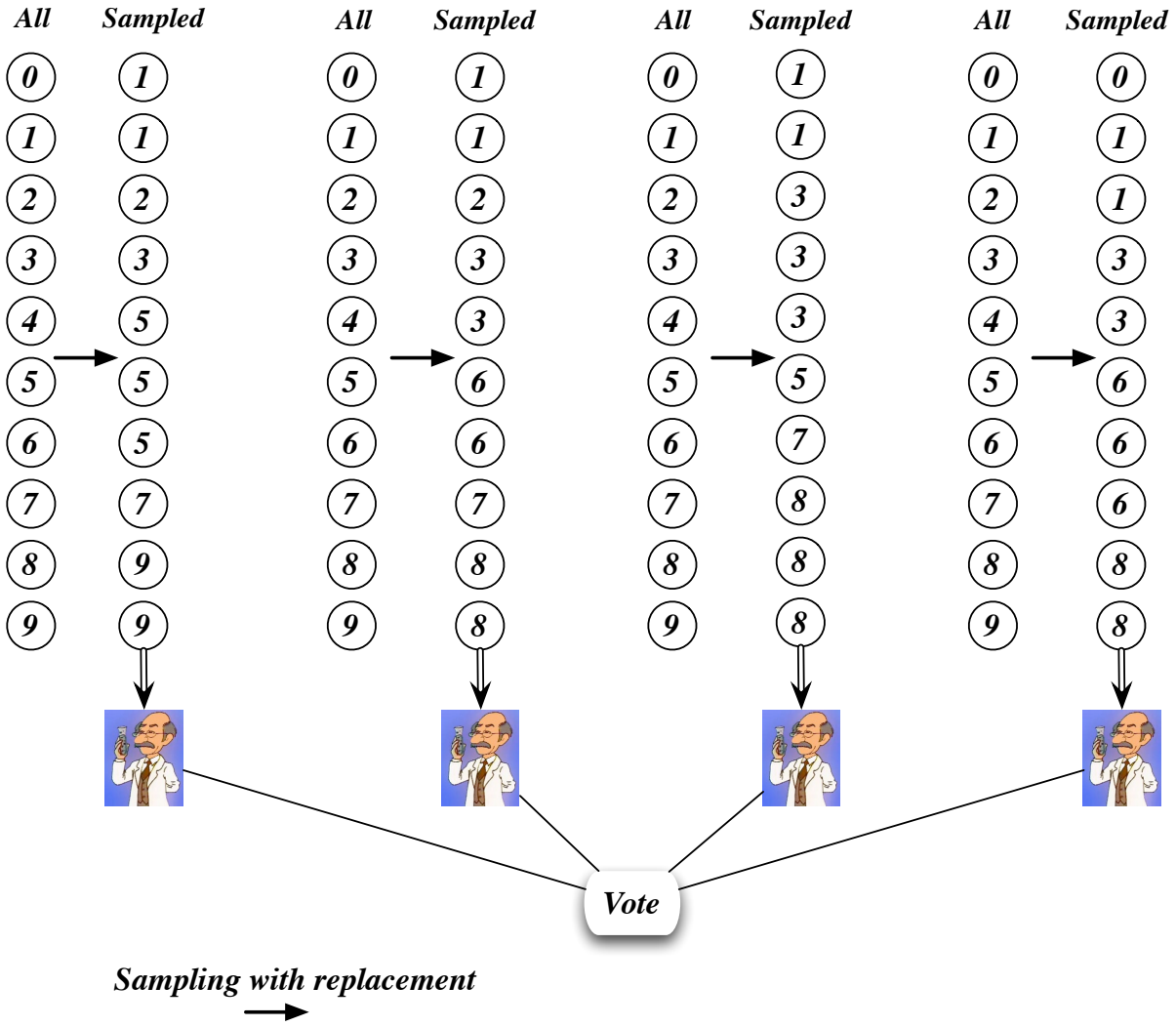
q_2 and q_4 are repeated; q_3 and others are missing.

Second Expert Sees A Different Sampling

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	12	1003	0.97	...	0.12
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_3	No	3	27	0.12	...	0.13
q_3	No	3	27	0.12	...	0.13
q_3	No	3	27	0.12	...	0.13
q_6	No	44	1212	0.29	...	0.42
q_8	Yes	78	42	0.44	...	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	12	3141	0.92	...	0.17

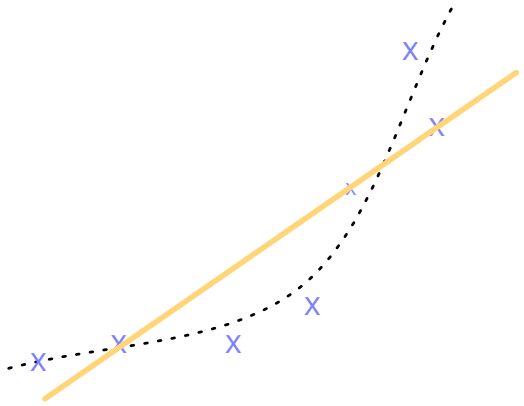
q_3 is repeated; q_4 and others are missing.

"Bagging" is the Formal Name for This Method

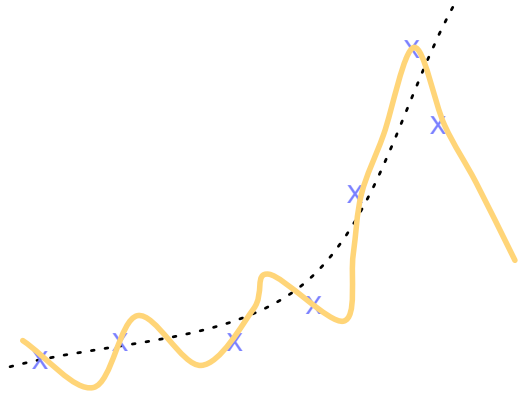


Why Do Ensembles Work? (A)

- A statistical model is a *noisy* model of reality.
- Bias error:
  Model too simple, underfits.
- Variance error:
  Model too complex, overfits.
- Bias/variance is a trade-off.
- Ensembles:
  - Use methods with low bias...
    but high variance ...
    and average to reduce variance!
- Result:
  low bias error *and* low variance error.
  No hand tuning needed.



Too simple a model underfits the data.



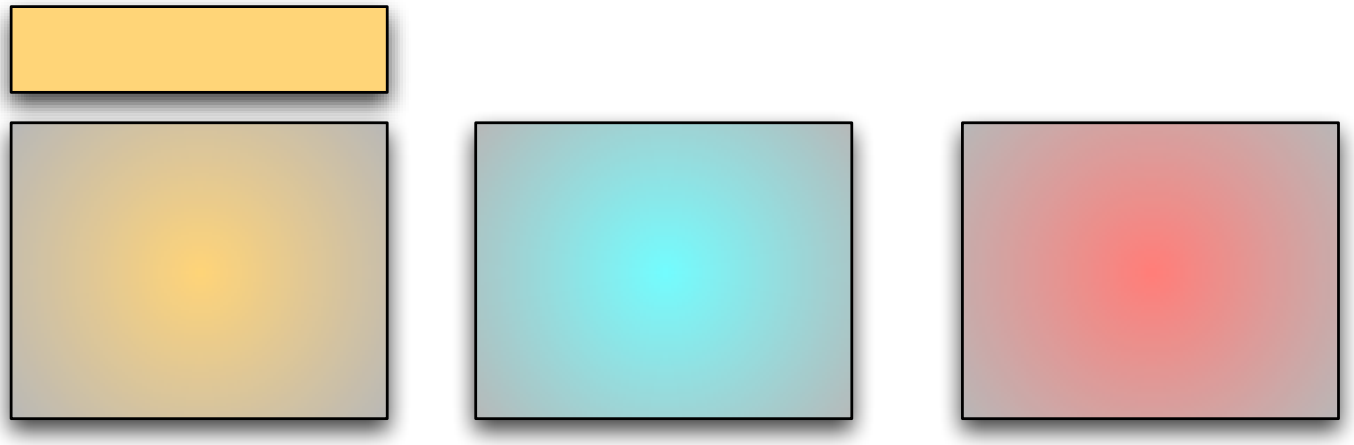
Too complex a model overfits the data.

Why Do Ensembles Work? (B)

One key is *diversity* [6].

Imagine: three classes, each expert only 10% accurate, and when wrong, chooses at random among the three classes.

Then the crowd of experts is perfectly, 100% accurate!



One group of unconfused experts amid the foggy error.

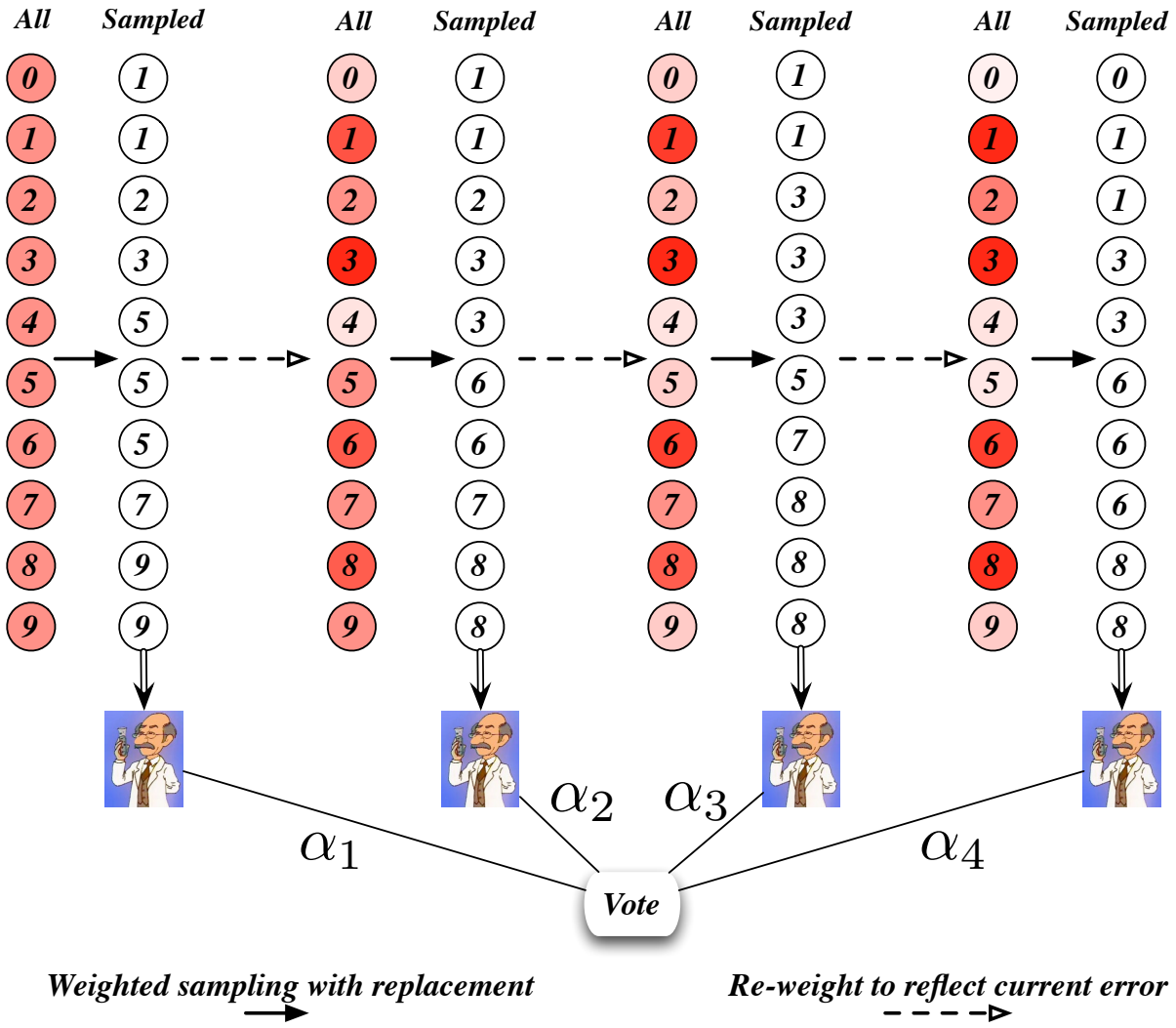
Note: diverse, *random* error is difficult to achieve[2].

Conclusions (version 1.1)

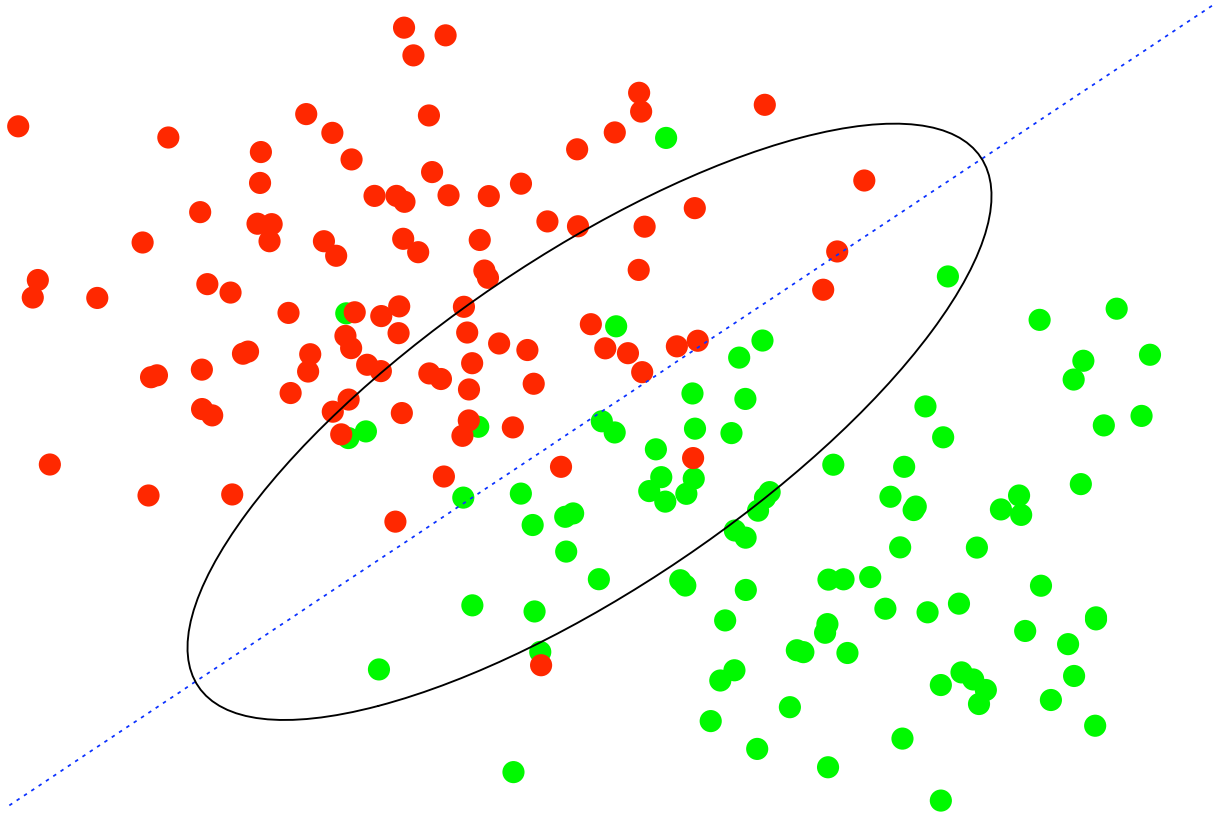
If you use supervised machine learning, then ...

- *Only* if you have clean training data (and you probably don't), use **ivoting**.
- Otherwise use bagging.

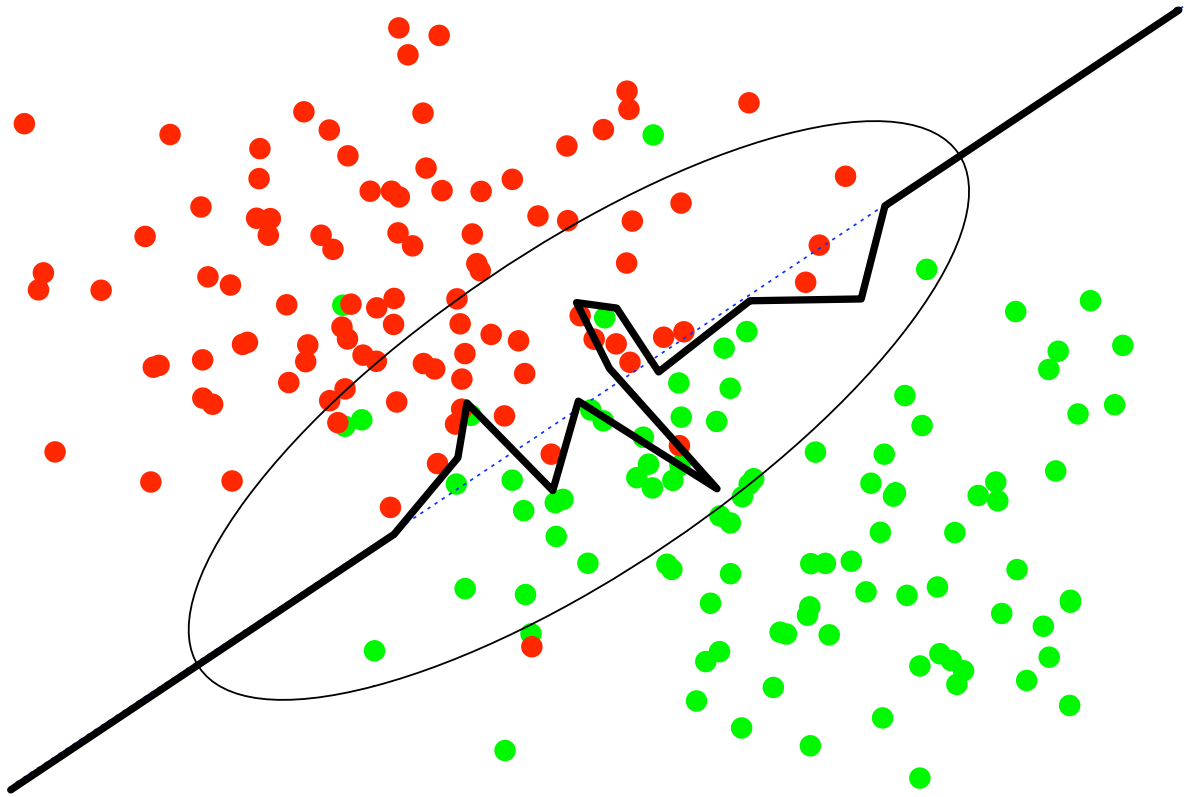
Boosting


All
Sampled
All
Sampled
All
Sampled
All
Sampled
α_1
α_2
α_3
α_4
Vote
Weighted sampling with replacement
Re-weight to reflect current error

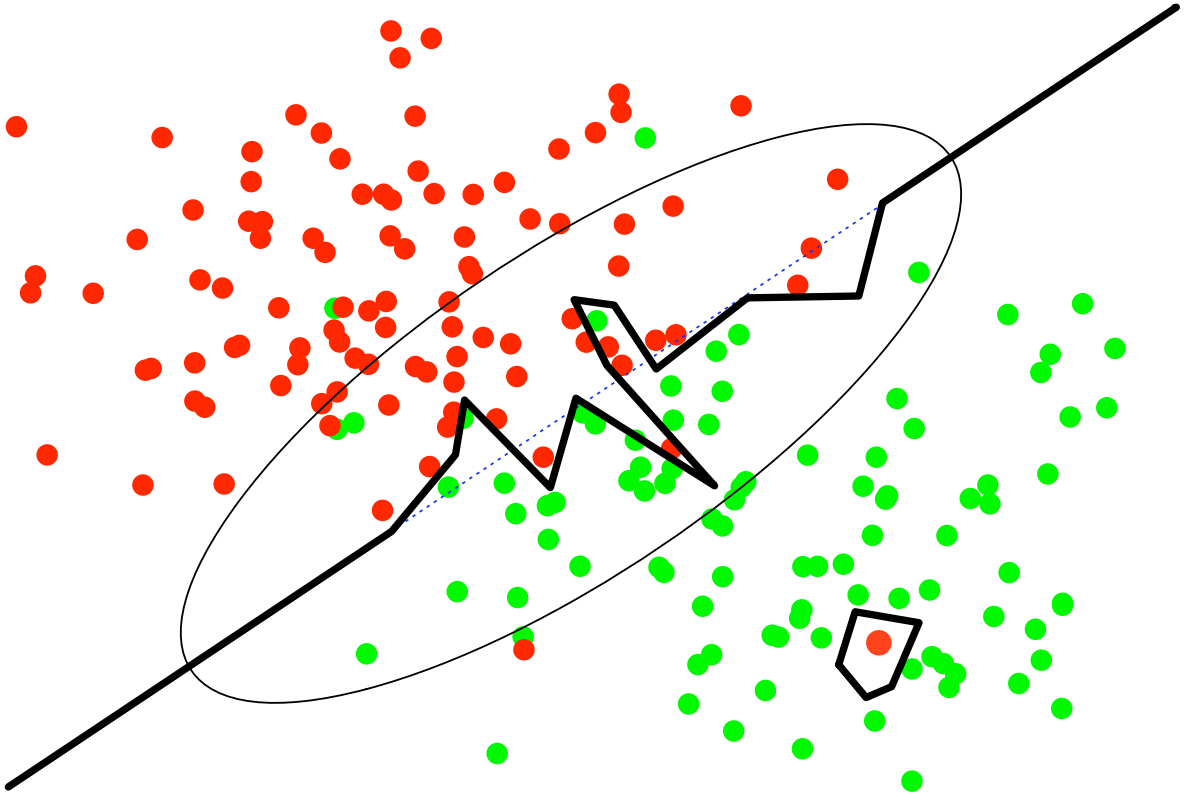
How Boosting Works



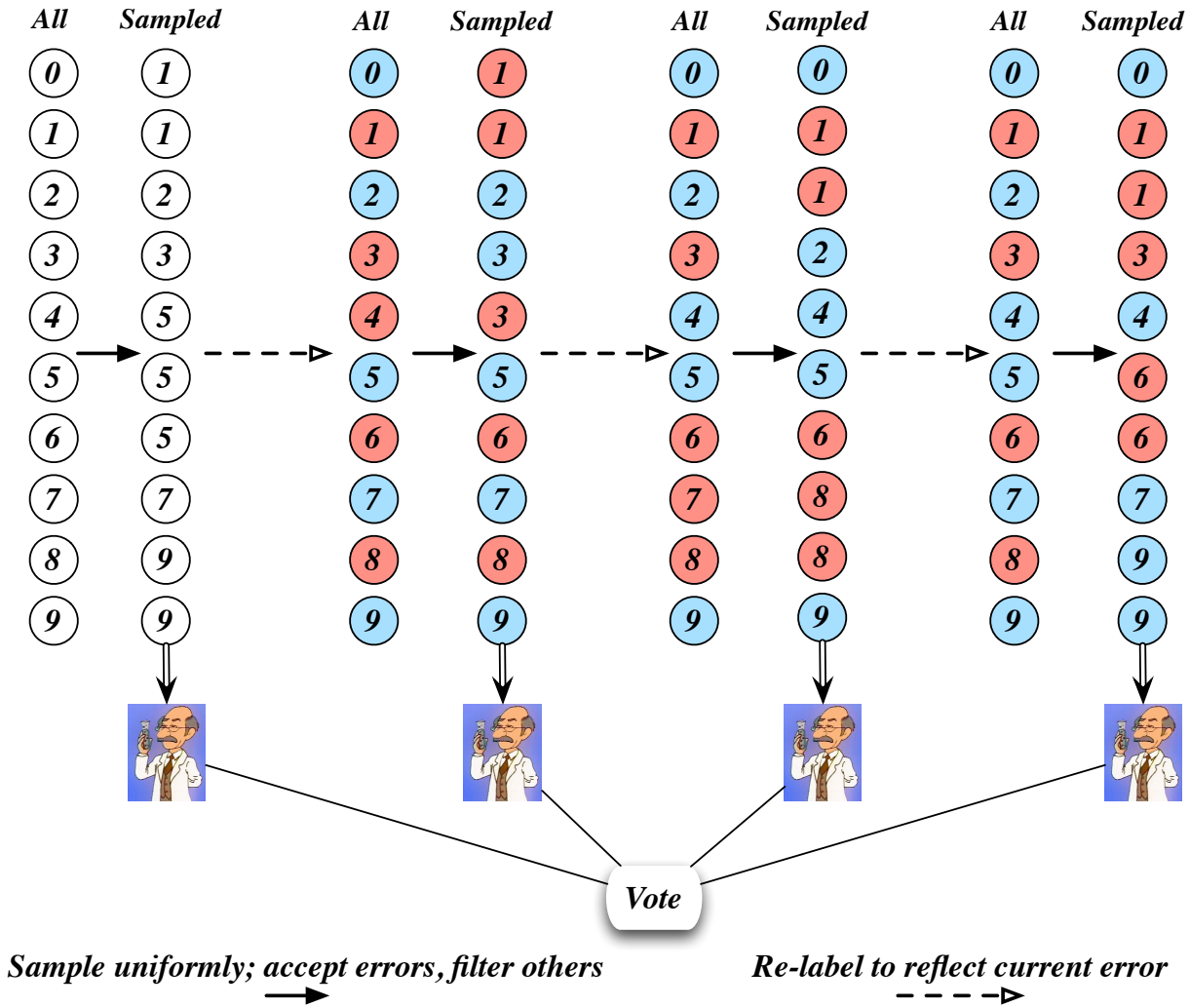
How Boosting Works Too Hard



Boosting Corrupted by Noise



Ivoting


All
Sampled
All
Sampled
All
Sampled
All
Sampled
Vote
Sample uniformly; accept errors, filter others
Re-label to reflect current error

Conclusions (version 1.2)

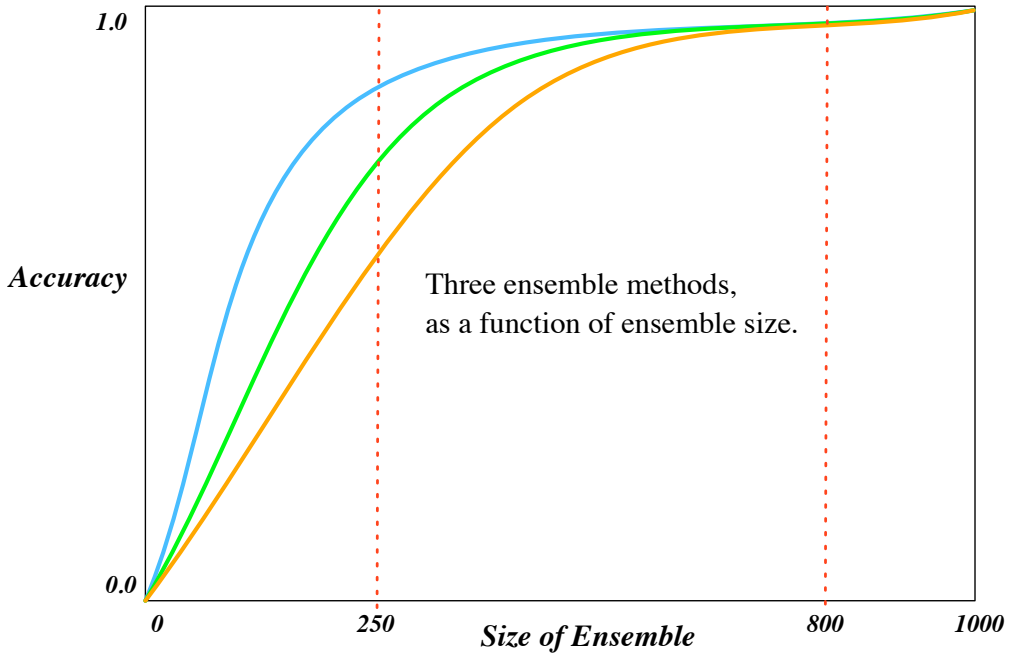
If you use supervised machine learning, then ...

- *Only* if you have clean training data (and you probably don't), use ivoting.
- Otherwise use bagging.

and use **out-of-bag (OOB)** validation to set ensemble size[3].

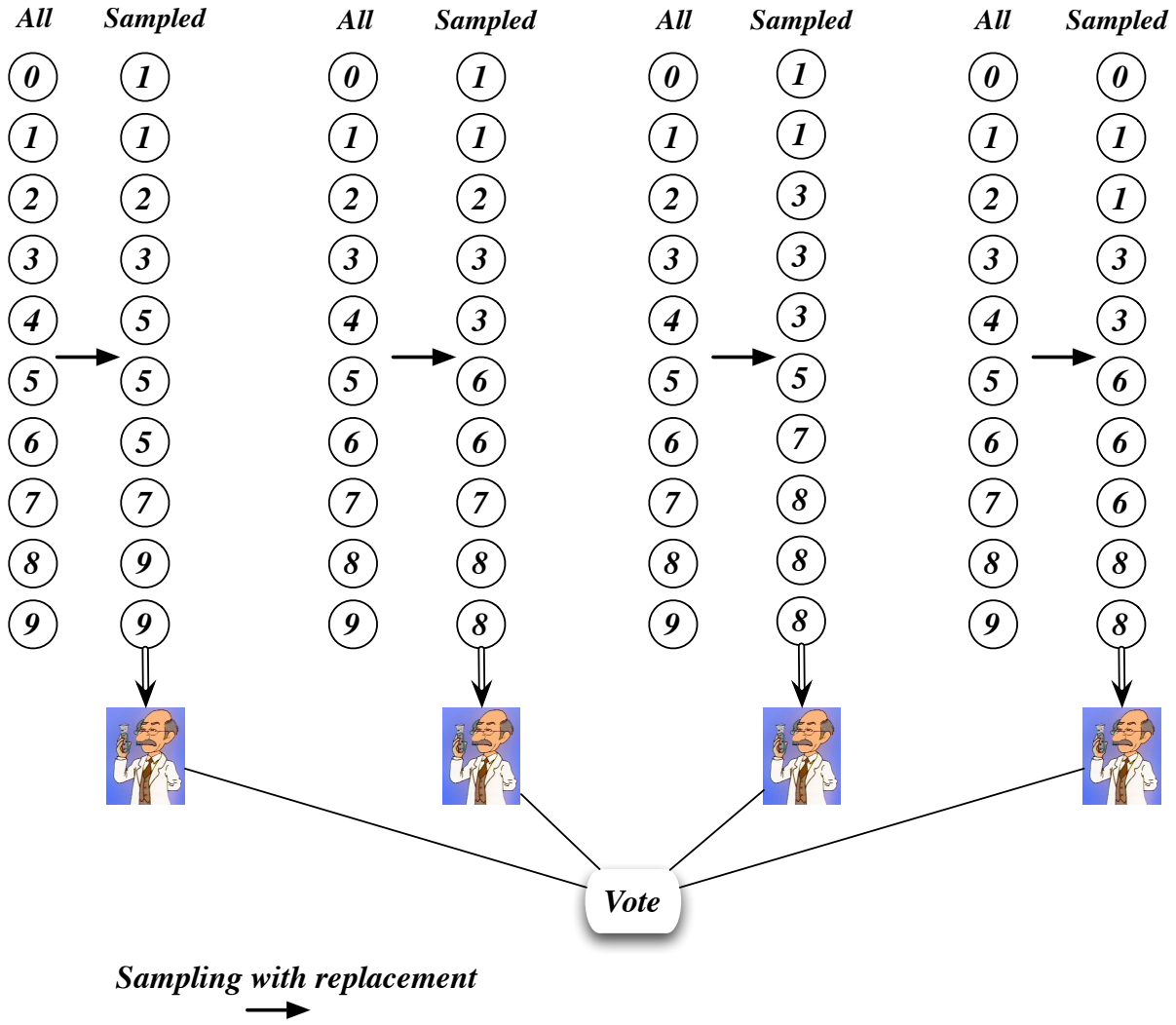
How Big An Ensemble Do You Need?

Don't use fixed size ensembles. They can be deceptive.
Instead, stop when accuracy levels off.



But how to measure accuracy? *Don't* just use the training data.
Use a separate validation set? Sure, but they are rare and costly.
Out-of-bag (OOB) validation is easy and cheap.

Every Classifier Doesn't See Some Samples

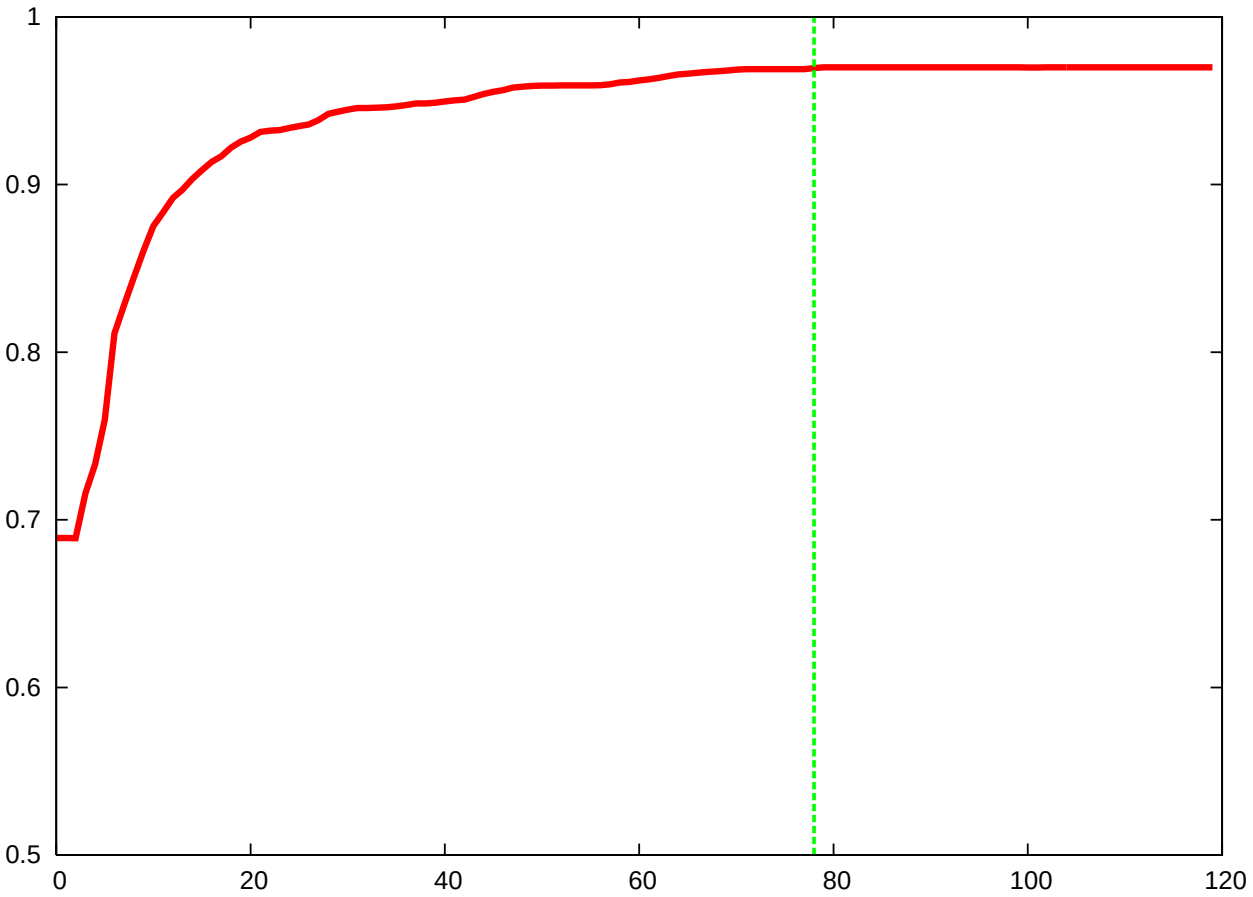


Every Sample Is Unseen by Some Classifiers!!

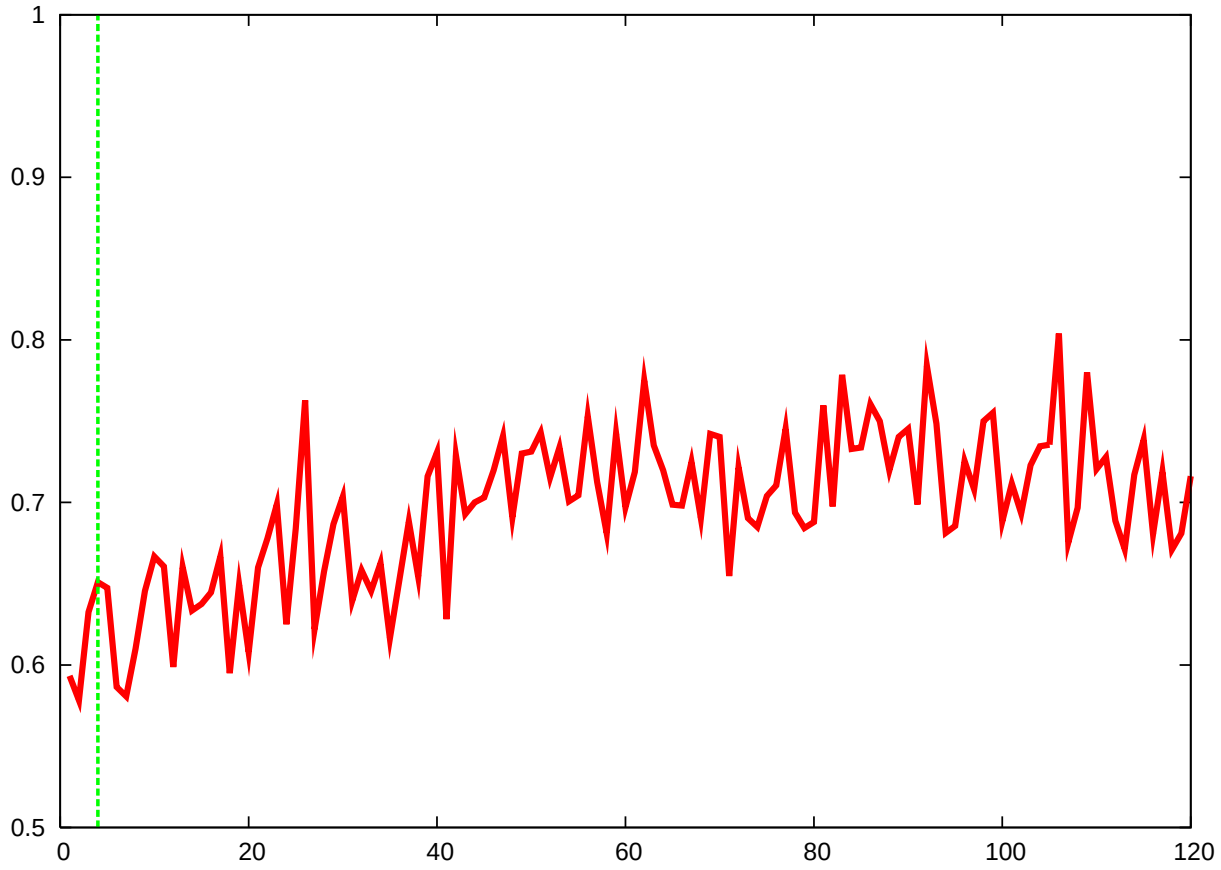
The classifiers that didn't see the sample can be fairly used to test it.

Sample 2 can be tested by E3 and E4; Sample 4 by E1, E2, E3 and E4.
Each sample can be tested by a substantial fraction of the classifiers.
So the over all accuracy is accumulated, one sample at a time.

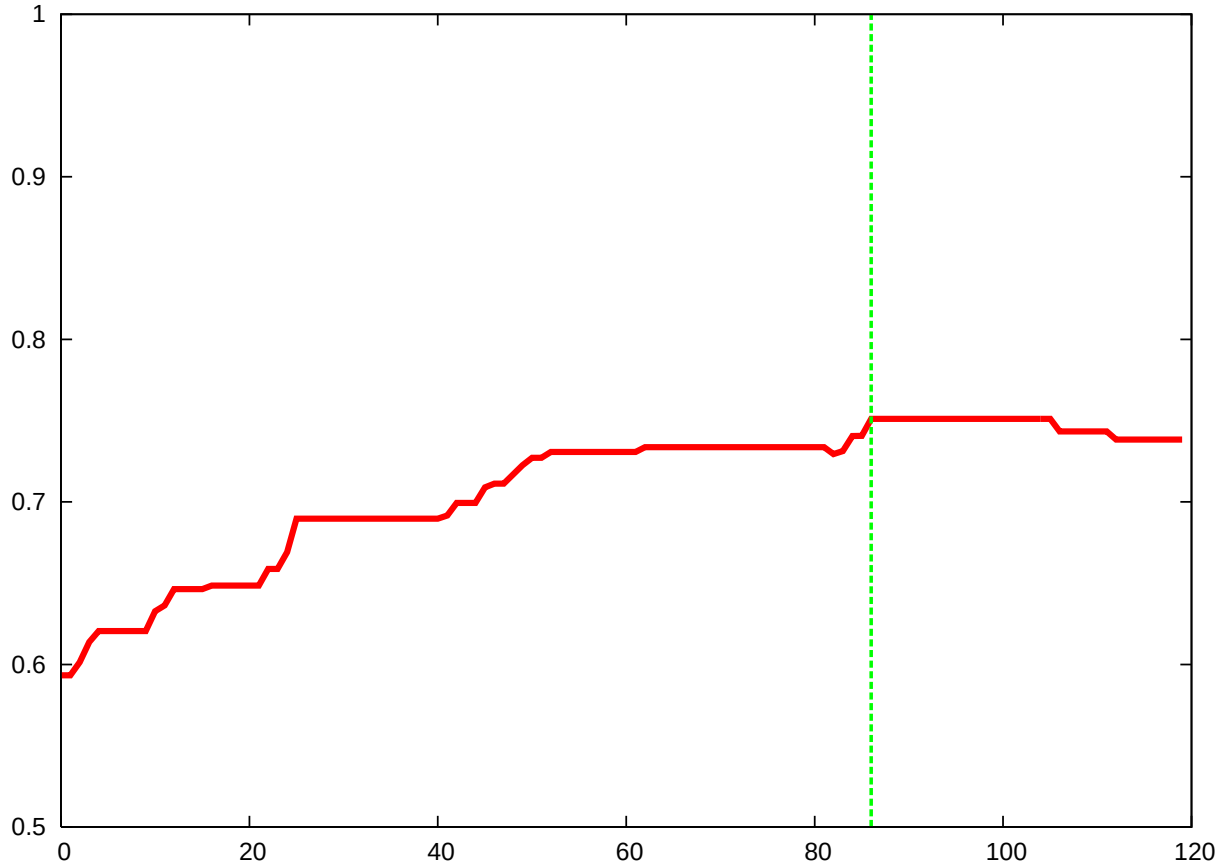
When To Stop? When Accuracy Flattens Out



Though That Can Be Tricky to Detect . . .



. . . Though We Have Our Ways.



$w_{\text{large}} = 20$

Conclusions (version 1.4)

If you use supervised machine learning, then ...

- *Only* if you have clean training data (and you probably don't), use ivoting.
- Otherwise ...
  - If you are using **unstable** base classifiers, use bagging.
  - If you are using **stable** base classifiers, use **small, optimized ensembles** or **random subspaces**.

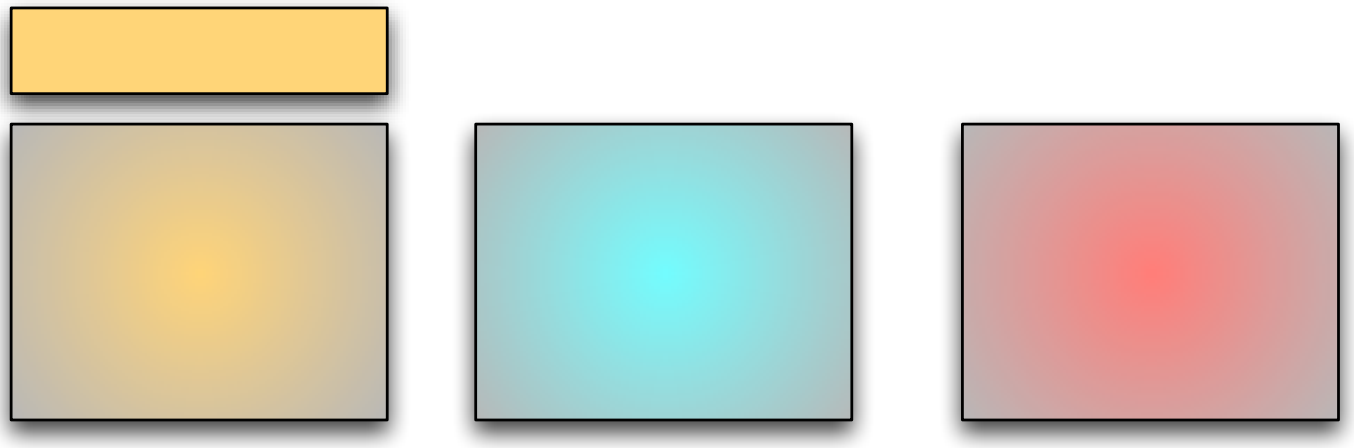
and use out-of-bag (OOB) validation to set ensemble size[3].

Reprise: We need Diverse Classifiers

One key is *diversity* [6].

Imagine: three classes, each expert only 10% accurate, and when wrong, chooses at random among the three classes.

Then the crowd of experts is perfectly, 100% accurate!

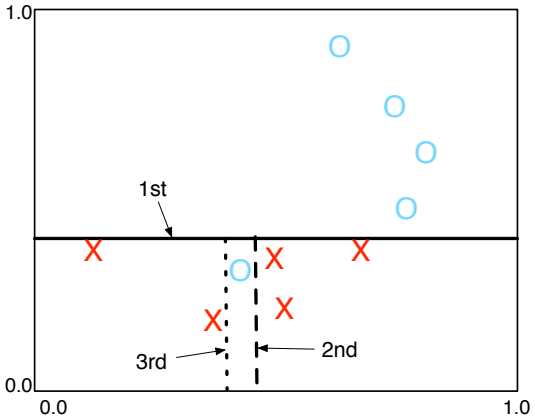


One group of unconfused experts amid the foggy error.

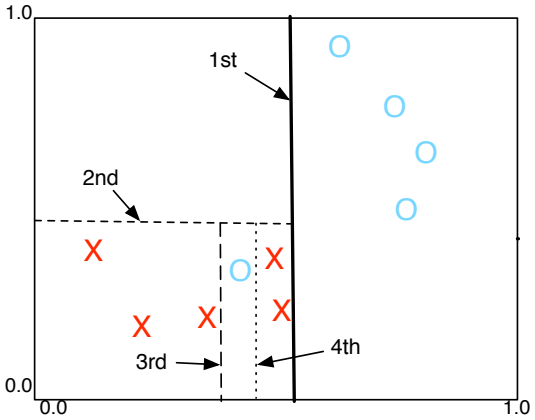
Note: diverse, *random* error is difficult to achieve[2].

Unstable Classifiers Are Easily Diverse

- "Unstable" is the same as "high variance error".
- Easier to get diverse classifiers from an unstable algorithm.
- Examples: decision trees, neural nets.



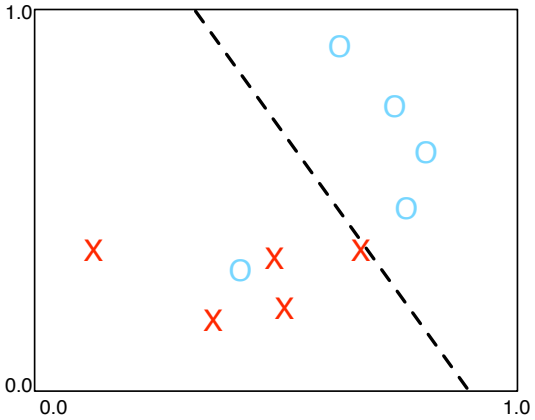
Small changes in sampling . . .



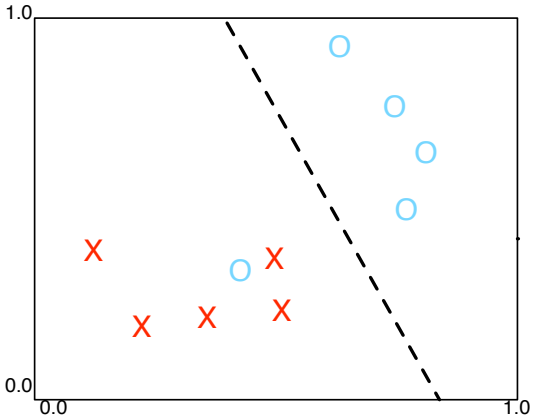
. . . make big changes in classifier.

Stable Classifiers Resist Diversity

- "Stable" is the same as "low variance error".
- Bagging won't pull diverse classifiers from an stable algorithm.
- Examples: naive Bayes, support vector machines (SVMs), conditional random fields (CRFs).



Small changes in sampling . . .



. . . make small changes in classifier.

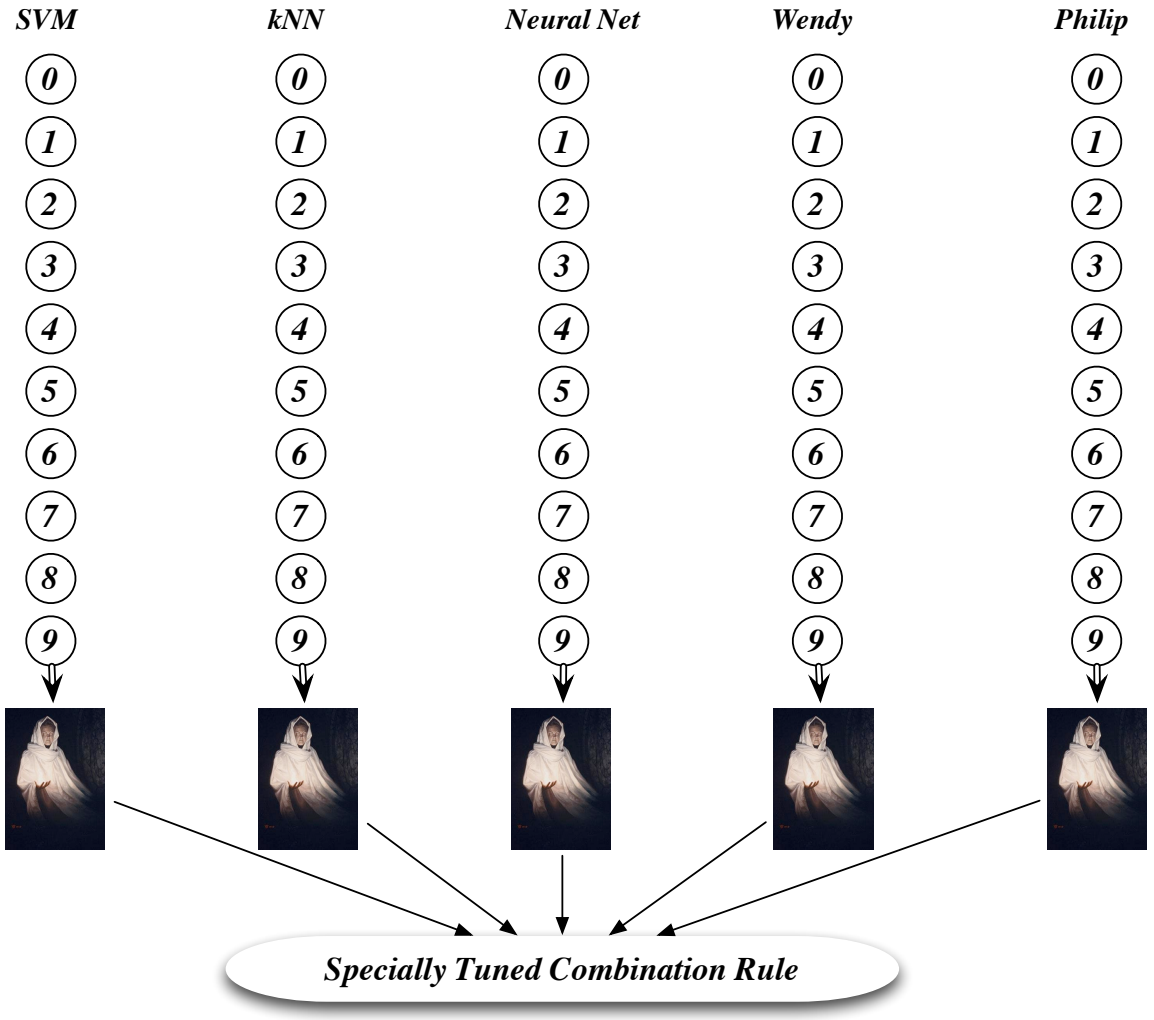
Reprise: Bagging Chopped Data by Rows

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_2	Yes	99	2	0.33	...	0.03
q_4	Yes	16	183	0.08	...	0.58
q_4	Yes	16	183	0.08	...	0.58
q_5	No	17	665	0.36	...	0.64
q_8	Yes	78	42	0.44	...	0.52
q_9	No	59	7012	0.37	...	0.23
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_{N-1}	Yes	36	1812	0.47	...	0.17

Random Subspaces Chops by Column

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	–	1003	–	...	0.12
q_2	Yes	–	2	–	...	0.03
q_3	No	–	27	–	...	0.13
q_4	Yes	–	183	–	...	0.58
q_5	No	–	665	–	...	0.64
q_6	No	–	1212	–	...	0.42
q_7	No	–	24	–	...	0.88
q_8	Yes	–	42	–	...	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	–	3141	–	...	0.17

Or: Tuned Ensembles of Strong Classifiers



Conclusions (version 1.6)

If you use supervised machine learning, then ...

- *Only* if you have clean training data (and you probably don't), use ivoting.
- Otherwise ...
  - If you are using unstable base classifiers, use bagging.
  - If you are using stable base classifiers, use small, optimized ensembles or random subspaces.
- If you have huge data, or distributed data, use **bozos**.

and use out-of-bag (OOB) validation to set ensemble size[3].

Ensembles From Tiny Subsamples

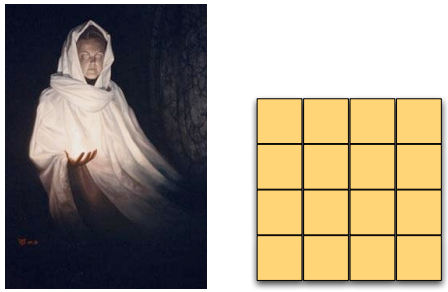
Traditional: Use 100% of training data to build a sage.

Ensembles: Use randomized 100% of training data to build an expert. Repeat to build many experts. Vote them.

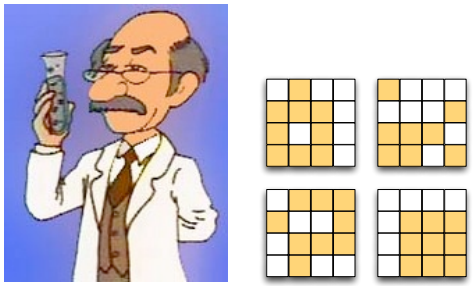
Sandia: Use a semi-random 1% of the training data to build a "bozo". Repeat to build very many bozos. Vote them.

The experts beat the sage[1].

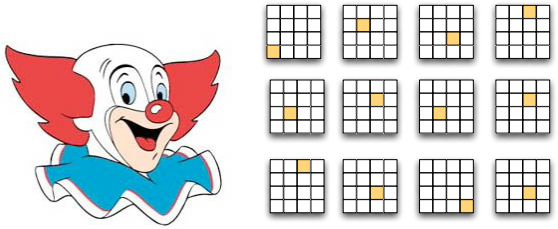
The bozos beat the experts[5].



Sage sees all the data.

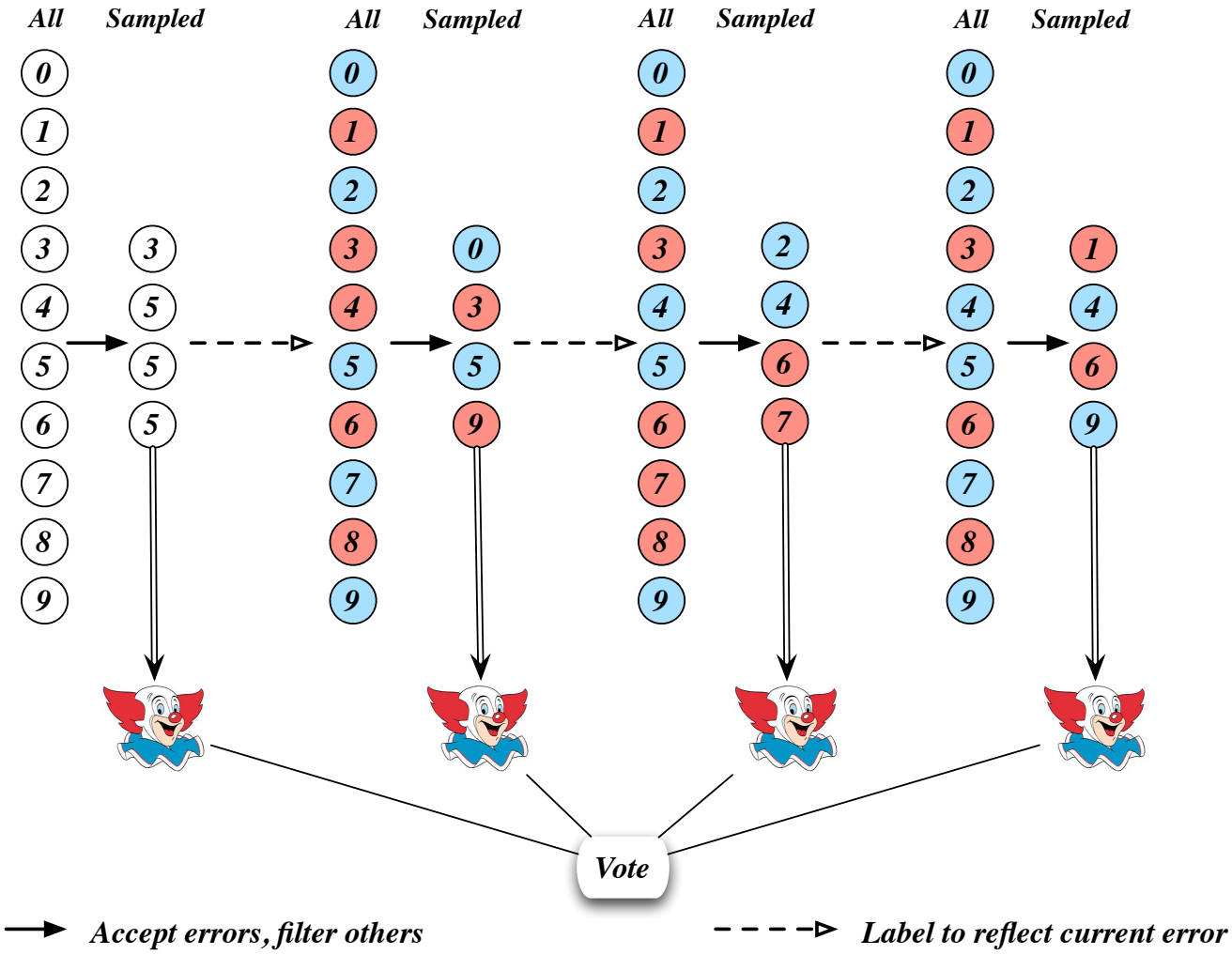


Each expert sees 2/3rds of the data.



Each bozo sees a tiny fraction.

Bozos: small data subsamples

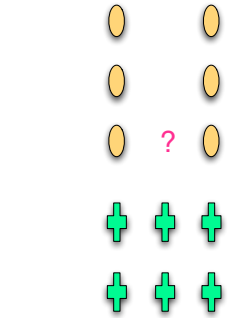


How To Get Started

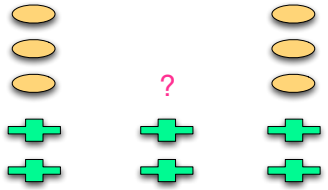
- Feel free to contact me: Philip Kegelmeyer, 925 294-3016, wpk@sandia.gov, Sandia National Labs, Livermore, CA.
- Prepare and format your training data; educational.
- Background reading: [7, 4, 9], and back proceedings of the "Multiple Classifier Systems" conferences.
- Evaluate methods correctly [3].
- Open source software: Weka, R and Rattle.
- My own AvatarTools, www.ca.sandia.gov/avatar (practical details and demo in a few slides)
- If starting from scratch, use **decision trees** and **random forests**.

Decision Trees Over Other Methods

- "No Free Lunch"[8] says the method doesn't matter . . .
  but only true for *clean* data!
- Most methods require an attribute distance metric . . .
  so attribute normalization matters.
- Decision trees don't need distance metric.
  - Use ordinal relations only.
  - Attributes need not be normalized.
  - Also, immune to noise attributes.
- With ensembles, no need to prune[5].



Unknown assigned differently . . .

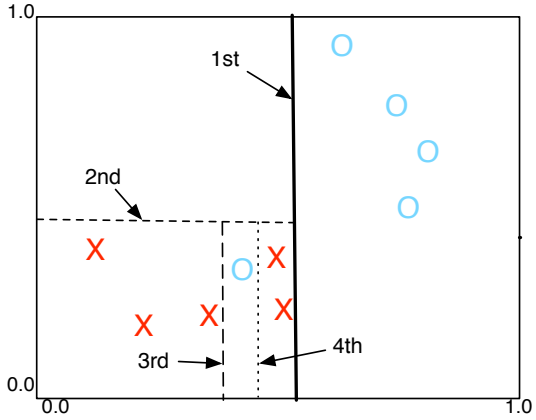


. . . depending on scaling

Decision Trees and Distance Metrics

- How to partition attribute space?
- For the current population:
  - Consider each attribute separately.
  - Consider each threshold for that attribute.
  - Pick attribute and threshold which "best decreases impurity".
  - Use them to partition the data into two child data sets.

  Repeat with each child.
- Best attribute and threshold is *independent* of scaling.
- Irrelevant attributes ignored in the presence of relevant attributes.



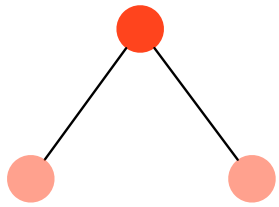
Attribute space partitioned.

Random Forests: Like Subspaces, But For Trees

Queries	Relevant?	PageRank	Fresh?	Unique?	...	Distinct?
	Truth	a_1	a_2	a_3	...	a_K
q_1	Yes	12	1003	0.97	...	0.12
q_2	Yes	99	2	0.33	...	0.03
q_3	No	3	27	0.12	...	0.13
q_4	Yes	16	183	0.08	...	0.58
q_5	No	17	665	0.36	...	0.64
q_6	No	44	1212	0.29	...	0.42
q_7	No	42	24	0.33	...	0.88
q_8	Yes	78	42	0.44	...	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	12	3141	0.92	...	0.17

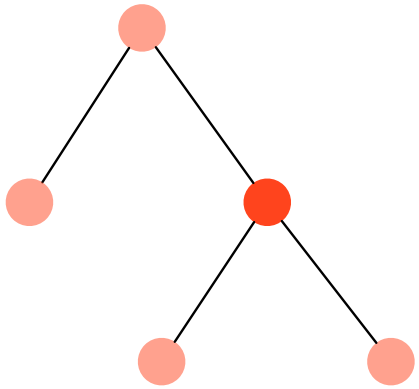
Use Different Attributes at *Each* Split

Queries	Relevant? Truth	Fresh? a_2	...	Distinct? a_K
q_1	Yes	1003	...	0.12
q_2	Yes	2	...	0.03
q_3	No	27	...	0.13
q_4	Yes	183	...	0.58
q_5	No	665	...	0.64
q_6	No	1212	...	0.42
q_7	No	24	...	0.88
q_8	Yes	42	...	0.52
$\vdots$	$\vdots$	$\vdots$		$\vdots$
q_N	No	3141	...	0.17



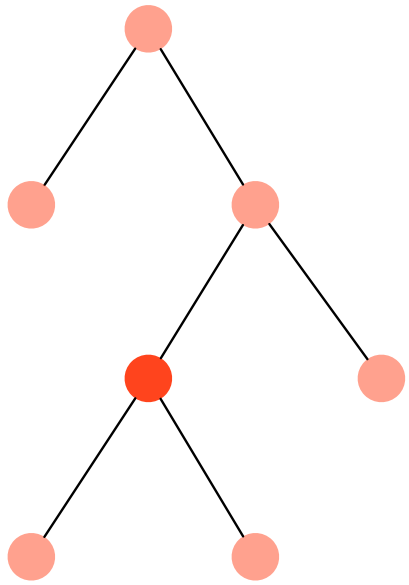
Use Different Attributes at *Each* Split

Queries	Relevant? Truth	PageRank a_1	Unique? a_3	...
q_1	Yes	12	0.97	...
q_2	Yes	99	0.33	...
q_3	No	3	0.12	...
q_4	Yes	16	0.08	...
q_5	No	17	0.36	...
q_6	No	44	0.29	...
q_7	No	42	0.33	...
q_8	Yes	78	0.44	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
q_N	No	12	0.92	...



Use Different Attributes at *Each* Split

Queries	Relevant? Truth	Fresh? a_2	Unique? a_3	...
q_1	Yes	1003	0.97	...
q_2	Yes	2	0.33	...
q_3	No	27	0.12	...
q_4	Yes	183	0.08	...
q_5	No	665	0.36	...
q_6	No	1212	0.29	...
q_7	No	24	0.33	...
q_8	Yes	42	0.44	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
q_N	No	3141	0.92	...



Why? If A is attributes, N is samples, then trees are $O(AN \log N)$

So random forests is the fastest decision tree algorithm.

Conclusions! (release version 2.0)

If you use supervised machine learning, then ...

- *Only* if you have clean training data (and you probably don't), use ivoting.
- Otherwise ...
  - If you are using unstable base classifiers, use bagging.
  - If you are using stable base classifiers, use small, optimized ensembles or random subspaces.
- If you have huge data, or distributed data, use bozos.
- If starting from scratch, use decision trees and random forests.

...and use out-of-bag (OOB) validation to set ensemble size[3].

Getting Access to AvatarTools

- Use the code on the ICC:
  - For $CLUS equal to tbird, shasta, spirit, or liberty:
    - Add `/projects/ascdd/avatar/$CLUS/current/bin` to PATH
    - Add `/projects/ascdd/avatar/$CLUS/current/man` to MANPATH
- Or build it yourself:
  - www.ca.sandia.gov/avatar
  - Standard Unix process; unpack tarball, configure, make.
  - Builds and passes tests on Mac, Linux, and Solaris.

Getting Started with AvatarTools

- See www.ca.sandia.gov/avatar for sample data and a tutorial. (And a video version of this talk.)
- Set up your data:
  - Make a comma separated foo.data file.
  - Start it with an optional "#labels" line.
  - Run data_inspector to create foo.names.
- Do analysis:
  - avatardt to train or test ensembles, or both.
  - mpirun avatarmpi to train in parallel.
  - crossvalfc to use cross-validation to assess accuracy.
  - rfFeatureValue to assess feature importance. (Warning: experimental.)

Pause for Demo?

A menagerie of AVATAR applications, past and current:

- Search by example in NW simulation data.
- Early detection of optics defects in the NIF beamlines (LLNL).
- Determine friend or foe from body movement.
- Detection of supernova in nightly scans (LBL).
- Word classification for entity extraction, for building graphs.
- Predict successful gene expression process parameters.
- Detecting and identifying "ideology" in documents.

References

[1] Banfield, R. E., Hall, L. O., Bowyer, K. W., Bhadoria, D., Kegelmeyer, W. P., and Eschrich, S. A comparison of ensemble creation techniques. In *Proceedings of the Fifth International Conference on Multiple Classifier Systems, MCS2004* (2004), J. K. F. Roli and T. Windeatt, Eds., vol. 3077 of *Lecture Notes in Computer Science*, Springer-Verlag.

[2] Banfield, R. E., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. Ensemble diversity measures and their application to thinning. *Information Fusion Journal 6*, 1 (March 2005), 49–62.

[3] Banfield, R. E., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (January 2007), 173–180.

[4] Breiman, L. Bagging predictors. *Machine Learning 24* (1996), 123–140.

[5] Chawla, N. V., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. Learning ensembles from bites: A scalable and accurate approach. *Journal of Machine Learning Research 5* (2004), 421–451.

[6] Condorcet, N. Essai sur l'application de l'analyse à la probabilitè des decisions rendues à la pluralite des voix. Correspondence, 1785. Paris.

[7] Dietterich, T. G. Ensemble methods in machine learning. *Lecture Notes in Computer Science 1857* (2000), 1–15.

[8] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification.* Wiley-Interscience Publication, 2000.

[9] Robert P.W. Duin, D. M. T. Experiments with classifier combining rules. In *Multiple Classifier Systems 2000* (2000), J. Kittler and F. Roli, Eds., no. 1857 in Lecture Notes in Computer Science, Springer-Verlag, pp. 16–29.

Supplemental Slides Follow

Sidebar: Machine vs Human Classification

Beware of hand-crafted analysis rules!

- Humans are great[a] at subtle judgments, but ...
- ... Humans are terrible at codifying them:
  - We don't really understand what we do.
  - And when we do, we don't describe it well.
- So human prediction rules tend to
  - be time-consuming to build,
  - overfit the data (and only the most recent data),
  - be brittle and in need of frequent tweaking.
- Better to ask: how can I turn this rule into an attribute?

[a] And, really, we're not so great, if statistics are involved, or rare events, or the need to consider more than seven factors at once, or ...

Sidebar: Use Human Rules for *Operations*, not Analysis

- Machine learning gives an object a label, nothing more.
- What do you do with that labeled object? That's operations.
- If operations are complicated, use a rule system to keep track.

```
if LABEL(d) is "defect"
    then
    if AREA(d) is < 3mm
        then add_to_watch_list(d)
    else if AREA(d) is < 5mm
        then send_alert(d)
    else push_panic_button(d)
```

- (Note: the line between analysis and operations can be fuzzy.)